Inquisitive Defect Cache: A Means of Combating Manufacturing Induced Process Variation

Avesta Sasan, *Student Member, IEEE*, Houman Homayoun, Ahmed M. Eltawil, *Member, IEEE*, and Fadi Kurdahi, *Fellow, IEEE*

***Abstract*—This paper proposes a new fault tolerant cache organization capable of dynamically mapping the in-use defective locations in a processor cache to an auxiliary parallel memory, creating a defect-free view of the cache for the processor. While voltage scaling has a super-linear effect on reducing power, it exponentially increases the defect rate in memory. The ability of the proposed cache organization to tolerate a large number of defects makes it a perfect candidate for voltage-scalable architectures, especially in smaller geometries where manufacturing induced process variation (MIPV) is expected to rapidly increase. The introduced fault tolerant architecture consumes little energy and area overhead, but enables the system to operate correctly and boosts the system performance close to a defect-free system. Power savings of over 40% is reported on standard benchmarks while the performance degradation is maintained below 1%.**

***Index Terms*—Cache, fault tolerance, low power, process variation, SRAM, voltage scaling.**

I. Introduction

USING minimum sized transistors to create compact SRAM structures, coupled with the analog nature of memory read/write operations, results in a considerably larger defect density in memories as compared to logic. In addition, it is shown in [1], [2], and [15] and will be further discussed in Section III that voltage scaling exponentially increases the memory cell failure probability. This implies that caches, which are the largest memory structures in the processor, abruptly limit the processor lower limit of supply voltage by introducing an exponential relation between their cell failure rate and the system supplied voltage. Since the failure probability under voltage scaling is exponential, fault tolerant techniques, such as row and column redundancy, that linearly increase the fault tolerance limit of the cache will fall short for significant reductions of the supply voltage (V_{cc}^{Min}) due to their limited fault coverage. This suggests that a fault tolerant mechanism, to be used in voltage scaled caches, should be able to tolerate a very large number of defects. The increase in fault coverage however should be achieved in a fashion such that: 1) the impact of fault handling mechanism on delay and performance of system is acceptable and 2) the total power consumption of the system including the new fault handling mechanism is lower than that of other simpler fault handling mechanism that operate at higher voltages handling lower number of defects with lower overhead.

In this paper, we explore the organization of a novel fault tolerant cache architecture which is capable of providing fault coverage for caches suffering from very high defect rate. We first introduced this organization in [3] for direct mapped caches with single word fetch per cache access. In this work, we further explore the usage of the inquisitive defect cache (IDC) to associative caches with multiword fetch groups, which resulted in significantly smaller area overhead. Furthermore, we introduce a mathematical bound on defect coverage of the proposed architecture relating the area and power of the proposed architecture to achievable (V_{cc}^{Min}). In [3], we investigated the fault coverage and voltage scalability of the system when a fixed frequency voltage scaling (FFVS) policy for scaling the cache supplied voltage is used. In this work, we extend our study and results examining the utilization of the proposed architecture for both voltage frequency scaling (VFS) and FFVS policies. Furthermore, the simulation environment (DINERO IV) presented in [3] has been replaced with SimpleScalar[1] to allow simulation of the architecture for well known SPEC2000 benchmarks. We have also discussed the alternative use of IDC, optimizing the system for fastest machine rather than the one with the lowest power. Finally we have elaborated on the effect of change in the distribution of threshold voltage on the system power and performance when equipped with IDC. This paper is organized as follows. Section II builds the necessary background and highlights the motivation behind this work. Section III explains the behavior of memory cell under voltage scaling exploring the effect of process variation in lower voltages. Section IV explains the effect of voltage scaling on memory logic. Section V introduces our proposed architecture. Section VI introduces an energy model to quantify the energy savings in lower voltages. Section VII presents a mathematical bound on IDC coverage. In Section VIII, we validate our proposed concept trough SimpleScalar simulation. Section IX explains how IDC could be used to realize a faster machine. Section X reviews the effect of change in threshold voltage distribution on system minimum voltage and energy consumption and finally, this paper is concluded in Section XI.

Manuscript received June 06, 2009; revised October 07, 2009; accepted June 01, 2010. Date of publication August 05, 2010; date of current version July 27, 2011. This work was supported in part by the National Institute of Justice (NIJ), Department of Justice (DOJ) under Grant 2006-IJ-CX-K044 and National Science Foundation (NSF) under Grant NSF CCF 0702797 and Grant NSF ECCS-0955157.

The authors are with the Department of Electrical Engineering and Computer Science, University of California at Irvine, Irvine, CA 92617 USA (e-mail: mmakhzan@uci.edu; hhomayou@uci.edu; aeltawil@uci.edu; aeltawil@uci.edu, kurdahi@uci.edu).

Digital Object Identifier 10.1109/TVLSI.2010.2055589

[1] [Online]. Available: http://www.simplescalar.com/

II. Prior Work

If a cache is defective, the system could still operate correctly provided that the faulty cache block or word is disabled. One way to achieve this is by marking a defective cache word/block with an extra bit that can be added to the set of flag bits of that block, conditioned that the added bit is not defective. In [5] this bit was referred to as the fault tolerance bit (*FT-bit*). The set of FT-bits for memory words or blocks is called the *defect map*. This defect map is used to turn a cache line off in case it is faulty. Turning a cache line off in an associative cache reduces the degree of associatively by one. In a direct mapped cache, on the other hand, every access to a disabled line will result in a miss. This scheme of disabling faulty blocks is done even when the cache is protected by single error correcting-double error detecting (SEC-DED) codes [6].

Replacement techniques, such as extra rows and columns, are also used in caches for yield enhancement [18] and for tolerating lifetime failures [7]–[10] With replacement techniques, there is no performance loss in caches with faults. However, the number of extra resources limits the number of faults that can be tolerated. Another form of redundancy is the use of extra bits per word to store an error correcting code. Sohi [7] investigated the application of a single error correcting and double error detecting (SEC-DED) Hamming code in an on-chip cache memory and found out that it significantly degrades the overall memory access time.

The work in [11] suggested resizable caches. This technique assumes that in cache layout, two or more blocks are laid in a single row. This assumption allows altering the column decoders to choose another block in the same row if a block is defective. In a technique called PADded caches [5] the decoder is modified to allow remapping of the defective blocks to other locations of the cache without destroying the temporal locality of data or instruction. A recent paper from Intel's Microprocessor Technology Lab [12] suggested the use of fault tolerant mechanisms trading off the cache capacity and associatively for fault tolerance. The proposed approaches while reducing the frequency allows scaling the voltage from a nominal 900 mv down to 500 mv in a 65-nm technology. The cache size is reduced to 75% or 50% depending on the mechanism that is used. In this scheme, data that is mapped to defective location is reallocated to healthy locations. The delay overhead, and the smaller cache size suggests that this scheme could not be used when high performance is desired, since, not only do we have to reduce the frequency, but also, due to smaller cache size, the number of cache misses will increase.

In this paper, a new architecture is presented specifically addressing power and yield improvement via fault tolerance. The ability to tolerate process and operational condition induced faults allows aggressive voltage reduction even in high performance mode (fixed frequency), which in turns leads to reduced power consumption while the performance is not degraded.

III. Effect of Voltage Scaling on Memory

A. Classification of Memory Errors

Memory cell failure is generally divided into ont of three distinct categories: 1) fixed errors or manufacturing defects, which are unaccounted shorts and opens in the memory cell and its supporting logic resulting in erroneous operation; 2) transient errors which are temporary memory malfunctions as a result of a transient situation such as an alpha particle attack; and 3) operating condition dependent errors due to manufacturing process variation in device physical parameters. This group of failures is not always present however with change in the memory operation conditions including voltage and temperature their number varies. Symptoms of these failures are change in cell access time, or unstable read/write operations to some locations

Significant contributors to the rapid increase in the number of process variation induced defects include line roughness, oxide thickness variations, as well as random dopant fluctuations (RDF) which results in mismatch between adjacent transistors threshold voltage (V_{th}) [1]. When applied to memory cells, the V_{th} variation results in large variation in access and write time to the memory cells. The dependence of V_{th} on temperature makes the write/access time also sensitive to die temperature. Furthermore, since the transistor speed is a strong nonlinear function of $V_{dd} - V_{th}$, the access/read time is also strongly and nonlinearly dependent on the supply voltage.

B. Modeling Memory Operations Time Under Process Variation

In this study, the following three types of memory failures were investigated including read failure (RF), write failure (WF), and access failure (AF).

1) ***Access failure*** occurs when in a given access time (from activation of wordline to activation of sense amplifier) a cell cannot create enough differential voltage between bitlines.
2) ***Write Failure*** occurs when in a given write time a cell cannot be written.
3) ***Read Failure*** is defined as a read operation that destroys the content of a cell.

We setup an experiment to study memory failures due to process variability. Adopting from [13], [20] due to RDF effect, threshold voltage variation is considered as the primary source of device mismatch. The circuit under test is a standard six transistor SRAM memory bit cell in a sub-bank of 64 cells connected to the same bitlines. The SPICE models used for the simulation were obtained from the Predictive Technology Model (PTM)[2] website in 32 nm. The $\sigma_{V_{th}}$ is expected to be $\sim$ 34 mV [21] in 32-nm technology. Monte Carlo simulations were performed with 10^7 iterations at each voltage level with a different threshold voltage assigned to each transistor in the SRAM cell based on a Guassian distribution. At each simulation point the following simple marching test is executed:

- ***write logic "1" to the cell (initialization);***
- ***write logic "0" to the cell:***
  - *measure time to write logic "0;"*
  - *finish SIM either on successful "write" or t = T_max_write.*
- ***Read a 0 from the cell:***
  - *measure time to create differential voltage Vdiff;*
  - *measure time to bit flip if happened;*

[2] [Online]. Available: http://www.eas.asu.edu/~ptm

- *finish SIM either on "bitflip" or t = T_max_read.*
- ***Write a 1 to the cell:***
    - *measure time to write logic "1;"*
    - *finish SIM either on successful "write" or t = T_max_write.*
- ***Read a 1 from the cell:***
    - *measure time to create differential voltage Vdiff;*
    - *measure time to bit flip if happened;*
    - *finish SIM either on "bitflip" or t = T_max_read.*

For each instant of Monte Carlo simulation the simulated access and write time at each voltage level is logged. Thereafter for each combination of voltage and access time cell failure probability is calculated by comparing the measured access time against defined access time. Please note that, the results will vary by changing the sizing of the cell, architecture of the memory bank, strength of the write transistors, or by using a different model card. The trend of change in the mean and standard deviation, and association of the failure rate to voltage and access time (as discussed in the next section) however will not change.

C. Defining System Cycle Time, Safety Margin (SM), and its Implications on Memory Cell Failure Rate

The failure probability curves are related to voltage frequency scaling policy. A voltage frequency scaling policy determines the memory cycle time (T_{cycle}) at each voltage point. T_{cycle} is the maximum of memory access time T_{Access} and memory write T_{Write}. For a nonpipelined cache the T_{Access} and T_{Write} are defined as the time required for complete read or write from memory, whereas for a pipelined cache the terms represent the longest pipeline stage during read and write operation. In order to account for process variation at each voltage T_{Access} and memory write T_{Write} should be accordingly larger than mean access time μ_T^{Acc} and mean write time μ_T^{Wri}

$$T_{\text{cycle}} = \mu_T^{\text{acc}} + T_{SM}^{\text{acc}} = \mu_T^{\text{wri}} + T_{SM}^{\text{wri}} \qquad (1)$$

where $T_{SM}^{\text{acc}} \& T_{SM}^{\text{wri}}$ are the Access and Write safety margins. By assigning different values to T_{cycle} we can define different VFS policies. Choosing a large T_{cycle} result in a large $T_{SM}^{\text{acc}} \& T_{SM}^{\text{wri}}$ and increased protection against process variation however degrading the system performance. On the other hand, a small cycle time reduces the safety margin and increases the probability of failure. Knowing the cycle time at each voltage point from the Monte Carlo simulation results of section C for each (voltage, frequency) a probability of failure could be extracted. The following equation allows us to define different VFS policies:

$$T_{\text{cycle}} = a.\max(\mu_T^{\text{acc}}, \mu_T^{\text{wri}}) + b.\max(\sigma_T^{\text{acc}}, \sigma_T^{\text{wri}}) + c. \qquad (2)$$

In this equation (a, b, c) are scale factors and $\sigma_T^{\text{Acc}} \& \sigma_T^{\text{wri}}$ are the standard deviation of access time and write time accordingly. At each voltage the associated mean and standard deviations at that voltage level is used. Equation (2) creates an expanded design space where different policies can be envisioned based on different values of (a, b, c)

If coefficients "a" and "b" are set to zero, a special case of VFS is created, which we refer to *fixed frequency voltage scaling (FFVS)*. In this policy when lowering the voltage, the cycle time stays constant. In this manner the increase in the mean access and write time the safety margin. This results in a rapidly increasing defect rate, albeit, maintaining the memory performance. The advantage of this type of voltage frequency scaling is that the cycle time is constant and if some fault tolerant mechanism could handle or mask the increase in the failure rate, a high performance system at lower power is obtained. Fig. 1 depicts the probability of failure for such a scenario. The probability of failure is the accumulated sum of all three types of failures (FR, RF, AF).

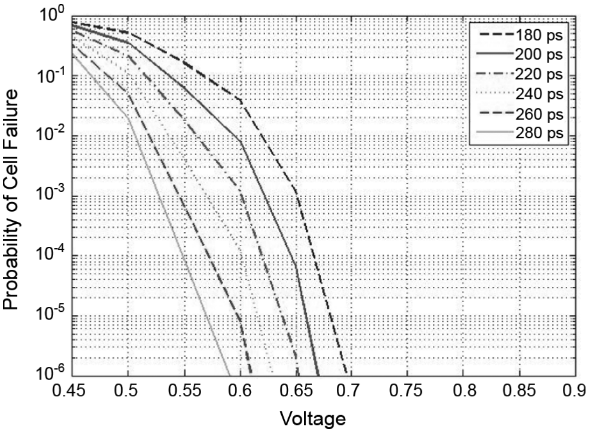

Fig. 1. Probability of cell failure for FFVS for different choices of c in (2).

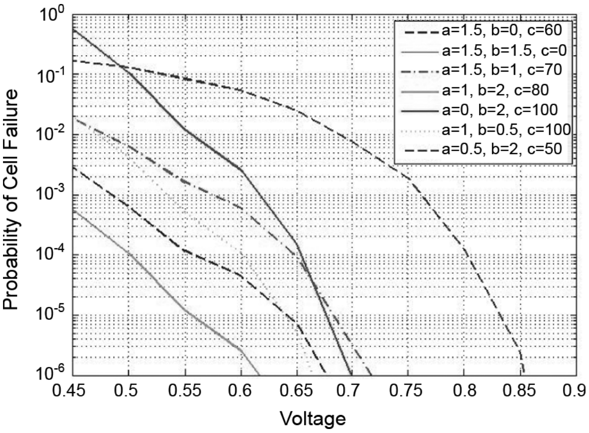

Fig. 2. Probability of cell failure for VFS for different choices of a, b, and c in (2).

The rate of change in the cell failure probability could be better controlled using a VFS policy. Fig. 2 illustrates how choosing different set of values (a, b, c) in (2) results in different trends in probability of failure curves.

Choosing a larger safety margin at a given voltages, increases the system's tolerance against process variation however the increased cycle time deteriorates the system performance. The implication on power consumption is twofold. On one hand, the lower voltage reduces the dynamic and leakage power consumption. On the other hand, the lower performance implies longer execution and the entire system leaks for a longer time. The appropriate lower voltage limit should be chosen by considering the maximum tolerable defect rate, the desired performance, and system total power consumption.

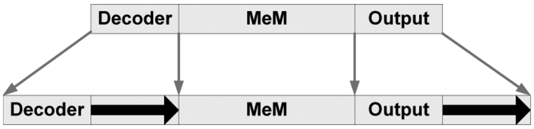

Fig. 3. Converting the access time of a nonpipelined cache to a pipelined cache.

TABLE I
NOMINAL TIMING OF A PIPELINE CACHE OF SIZE 16_{KB} WITH 1024 SETS, ASSOCIATIVITY OF 4 AND BLOCK SIZE OF 16 Bytes IN 32 nm

Data Delay (ps)		Tag Delay (ps)		Stage	Delay(ps)
decode	13	decode	17	1	17
wordline	16	wordline	14	2	38
Bit-line	22	Bit-line	21		
sense amp	7	Sense amp	6	3	18
Output	15	Output	18		

TABLE II
TERMS AND VARIABLES USED

E_{BLB} =	Energy per BLB read.
E_{Rc} =	Energy consumed to read cache
E_{Ridc} =	Energy consumed to read IDC
E_{Dc} =	Energy to decode cache
E_{Didc} =	Energy to decode IDC
V_{dd} =	Scaled memory supply voltage
V_{dd0} =	Nominal memory supply voltage
L =	Size of the Window of Execution
A =	Associability in the WoE
S =	Associability of the IDC cache
D =	Size of the IDC cache
$P_{f=}$	Probability of Cell failure
W =	Number of Words in each fetch group
T =	Number of bits in each word
P_{clean}^{access} =	Probability of clean access

IV. EFFECT OF VOLTAGE SCALING ON MEMORY LOGIC

In this section, we focus on a three stage pipeline cache as a case study, but note that the same arguments could be applied to wave pipelined and nonpipelined caches as well. We specifically choose the three stage pipeline cache to show some of the design considerations when a system is designed for FFVS or VFS policy.[3]

In a three-stage pipeline cache, the pipeline stages are: 1) decoding; 2) memory cell access (which is considered from activation of the wordline, to the analog operation of reading the target memory cells into bitlines and ends with sense amplifier activation); and 3) sensing, MUXing, and driving the output signals. In this configuration, the second stage (reading the memory cells) is the longest pipeline stage. This stage is analog and cannot be pipelined into further stages [1]. Fig. 3 illustrates the timing of a three stage pipelined cache operating at nominal V_{dd}. When pipelined, every stage in the cache has one cycle to complete its operation. The cycle length is determined by the delay of the longest stage in pipeline.

In an FFVS system the second stage delay stays constant throughout the voltage scaling and the shift in the distribution of the access/write time is translated to higher failure rates. One determinant for the lowest possible voltage $V_{cc}^{\min}$ is the tolerable failure rate. Another consideration for such system is the design of other stages of the pipeline (first and third stage in this case). These stages are purely logic stages. Reducing the voltage increases the logic propagation time and therefore reduces the operational speed of these stages. The delay of these stages, however, cannot exceed the delay of the second stage at lower voltages. Therefore, FFVS system must be designed such that all logic operations complete correctly at the lowest voltage. This means potentially using larger and faster transistors in the design. The delay of the logic [19] in the first and third stages is related to the supply voltage by the following equation:

$$t_{\mathrm{delay}} \sim \frac{V_{dd}}{(V_{dd} - V_{th})^{1.3}}. \tag{3}$$

On the other hand, if a VFS policy is used; as the voltage is lowered, the second stage delay (Δ) increases. Since all the stages of the pipeline should take equal time, the delay of the second stage is multiplied by the number of stages in the pipeline, increasing the access time by $(N.\Delta)$ and the cycle time by Δ. In such a case, the delay slack in the first and third pipeline stages allows the usage of smaller transistor in those stages to achieve lower power consumption. Table I depicts the delay of various stages in a 16_{KB} cache. The delays are obtained by simulating a post-layout realization of a 16_{KB} cache whose parameters are described in the same table. Date presented in Table III includes the delay of the decoder, memory cell, and the output logic for both tag and data. In this paper, we assume that all memory components are controlled by a single variable supply voltage. This assumption is made for the sake of simplicity and implementation purposes. The first and third stages have a lower limit on how much their voltage could be reduced before the stage delay exceeds the second stage delay (which defines the cache cycle time). In the proposed cache, when reducing the voltage, the tag decoding stage delay reaches the cycle time limit before other logic stages and since all stages are connected to a single supply voltage, this voltage defines the lower limit of supply scaling V_{cc}^{Min}

V. PROPOSED ARCHITECTURE

Fig. 4 illustrates our purposed architecture. A cache in this model is augmented with two auxiliary blocks. The bit lock block (BLB) is a defect map for the cache and the inquisitive defect cache (IDC) which is a small auxiliary cache for the purpose of defect tolerance. Both of these auxiliary structures are kept at high supply voltage when the voltage of the main cache is scaled. This is to guaranty reliable storage of defect information in the defect map. The size of these two structures compared to cache is much smaller and it will be shown that their augmented

[3] Although FFVS could be considered as a special case of VFS however the design implications—especially in pipeline structures—warrant considering FFVS separately.

TABLE III
SIMPLESCALAR SIMULATION SETUP

Variable	Value
Issue/decode	*4 words*
Branch prediction	*Combination*
ROB/LSQ	*128/64 entry*
L1 data cache	*16KB 3cycle pipelined, 4bank, 4 ways*
L1 inst cache	*16KB 3cycle pipelined, 4bank, 4 ways*
L2 cache	*Unified, 1M, 4way, 16cycle, 4 way*
Memory access latency	*(400 +8 per 8bytes) cycles*
L1 IDC	16 rows, 2ways
L2 IDC	64 rows,2 ways

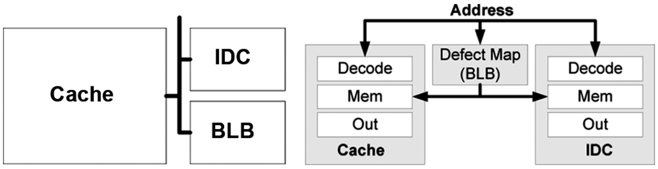

Fig. 4. Proposed architecture.

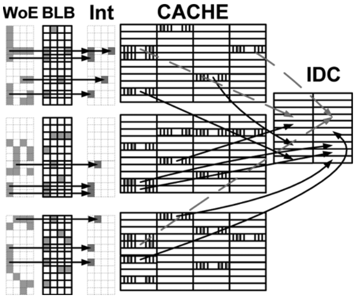

Fig. 5. In the first pass defective FGs in window of execution are mapped to IDC.

power penalty is much smaller than that saved by allowing aggressive voltage scaling within the main cache.

A. BLB Defect Map

The number of words that are read from a cache at each read cycle is referred to as a fetch group (FG). In this architecture, to create a defect map, we use one bit per FG in cache, which is stored in the BLB. The size of the BLB is therefore fixed and proportional to the size of cache.

B. Generating and Updating the BLB

As part of the operation of the system, we need to have a mechanism for generating the BLB and another one for updating it. Generating the BLB could be accomplished at manufacturing time, however testing for multiple vectors significantly increases the chip cost. Another approach is to generate the defect map at boot time. In this method, at manufacturing time, the memory is only tested for manufacturing defects at the nominal voltage process corners, fixing such errors using the available redundancy and leaving the process variation induced defects to the combination of the fault tolerant memory (FTM) and built-in self test (BIST) unit. At run time BIST tests the memory arrays at lower voltages and write the defect map into BLB. BIST generates a different BLB for operation at each PVT corner. Defect maps for previously tested voltages could be saved by the system in non volatile memory to avoid running the BIST in the future or can be rerun on demand.

BLB should also be updated by possible defects in the result of operation environment changes. (e.g., temperate) This is achieved in one or a combination of the following ways.

1) *Usage of Parity Bits:* Every time that a new error is discovered, its location is registered in a very small memory (in our case 16 or 32 entry). Since the newly discovered defect could be a soft error, BLB is not updated immediately. If the defective FG is detected again the location is tested using a marching test and upon confirmation it is considered as an operation environment induced error and is registered in BLB. Each BLB row has a dirty bit which is set if a bit in that row is updated. The presence of the dirty bit requires a write back of the BLB information when changing the voltage level.
2) Running the built-in BIST engine periodically.
3) Using dynamic temperature sensing (DTS) infrastructure, an approach which is already being adopted by many manufacturing companies in 65 nm and smaller geometries. DTS is geared towards avoiding overheating but can also be used for other purposes such as modifying the policy stacks. In this approach, the system senses the cache temperature. When a temperature increase exceeding a threshold is detected, the BIST engine is instructed to test the heated locations for temperature induced defects. Upon discovery of a defect, the BLB is updated and its dirty bit is set. Since temperature is a very slow changing variable such BIST updates will not occur frequently.

Although IDC is designed for dealing with process variation defects, it could also be used to deal with manufacturing defects in upper level memories (caches). In other words, manufacturing defects could be considered as process variation defects that are present at all corners of the PVT space. With this in mind if at the manufacturing the number of available redundancy is less than the number of manufacturing defects, conditioned that their number is smaller than available space in IDC and they are spatially distributed in FGs that are not in close proximity, they could be masked by the proposed mechanism allowing increase in the production yield.

C. Inquisitive Defect Cache Memory

The inquisitive defect cache (IDC) is a small associative auxiliary cache which temporarily holds FGs mapped to defective locations in the window of execution (WoE) of its associated cache. The conceptual view of an IDC enabled cache structure is illustrated in Fig. 5.

The size of IDC cache could be much smaller than the total number of defective FGs in a cache. In a sense, the IDC could be considered as a group of redundancy rows with the main difference being that mapping to redundancy is fixed and one time, whereas mapping to the IDC is dynamic and changes with changes in the addressing behavior of the program. Furthermore, in the IDC case, the defective FGs within the WoE of the program in the cache are mapped to IDC allowing much higher coverage than that of a redundancy scheme.

In Fig. 5, IDC, Cache, and BLB represent the Inquisitive defect cache, cache, and defect map (BLB), respectively. The WoE and "Int" are conceptual structures (do not exist but drawn to ease the understanding of IDC structure and its operation) representing the WoE of the program in the cache and intersection of the WoE and BLB, respectively. The BLB marks the defective FGs in the cache. The WoE marks the in-use section of the cache (FGs that the processor has recently visited). The "Int" is in the results defective FGs in the WoE which should be mapped to IDC. In Fig. 5, the IDC and cache have associativity of 2 and 4, respectively.

If the IDC size and associatively are chosen properly, the WoE of a process in the cache (after its first pass) should experience very few defective/disabled words and could be virtually viewed as a defect free segment in the cache by the processor.

If a FG is defective, its information could only be found in the IDC since every read/write access to defective words is mapped to the IDC. The BLB should be designed such that its access time is smaller than the cache cycle time. Considering the small size of the defect map when one bit per FG is chosen, and also the fact that the defect map voltage will not be scaled, makes it fairly easy to meet this design constraint.

On each access, all three components (BLB, IDC, and Cache) are accessed simultaneously. However immediately after BLB feedback the operation on either cache or IDC for the next cycle is gated to save power. The following section analyzes the access scenarios and derives models for energy consumption for each scenario.

VI. Energy Models

A. Dynamic Energy

An access to the cache, as illustrated in Fig. 6, initiates parallel access to cache, IDC & BLB. In this model however after the first cycle, based on defect map feedback, access to either cache or IDC is gated. The dynamic energy consumption when IDC or cache is gated is presented in

$$E = (E_{\text{BLB}} + E_{\text{Didc}}) + (E_{Rc}) * \frac{v_{dd}^2}{v_{\text{ddo}}^2} \tag{4}$$

$$E = (E_{\text{BLB}} + E_{\text{Ridc}}) + (E_{Dc}) * \frac{v_{dd}^2}{v_{\text{ddo}}^2}. \tag{5}$$

As a program shifts its WoE to a new location in cache, it may access defective locations not covered by IDC. In response cache declares a miss and data is retrieved from lower levels of memory hierarchy and placed in IDC. A cache miss due to access to a defective location in the cache which is not covered in IDC is referred to as "*soft miss*." A soft miss could also occur if the number of defective FGs that are mapped to one physical row in the IDC exceed the associatively of the IDC (this case is illustrated using dashed lines in Fig. 5).

Dynamic power is affected by the miss and soft miss rate. As the miss rate increases, the dynamic power consumption. High miss rate could cause the processor to stall, increasing the execution time. This in turn means that the processor and memory would leak for a longer time, increasing the total energy consumption. Considering that in smaller geometries leakage energy is comparable and even predicted to be dominant compared to dynamic energy, a considerable energy overhead is expected.

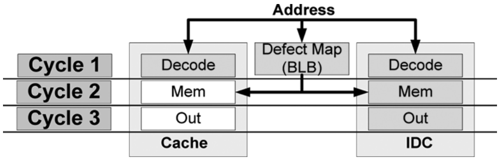

Fig. 6. Cache access scenarios when defect map refer to IDC, and access to cache is gated.

Consider a cache access scenario that results in a miss: the address is sent to BLB, IDC, and Cache. All three components start decoding the address at the same time. Following one of the two cases explained in Section VI, a miss will occur. System then initializes access to lower level memory. When data is ready in the lower level cache, it is transferred to upper level either by writing to IDC and/or Cache and the cache request is reinitiated. At this time the BLB, IDC, and Cache are accessed again in parallel.

B. Leakage Energy

Memory leakage power consumption is exponentially related to the memory supply voltage. At process technologies of 65 nm and below, the dominant leakage components are subthreshold leakage and gate leakage. The gate leakage with usage of high k material will be reduced by a factor of $\sim$ 100x Therefore we only consider the threshold voltage. The sub-threshold leakage in MOSFET transistors relates to voltage by

$$I_{\text{leak}} = \mu_0 \cdot \mathcal{C}_{ox} \cdot \frac{W}{L} \cdot e^{b\cdot(V_{dd}-V_{\text{ddo}})} \cdot v_t^2 \cdot (1-e^{-V_{ds}/v_t}) \cdot e^{-\Delta V_{th}/n\cdot v_t}. \tag{6}$$

In this equation, $v_t = KT/a$ is the thermal voltage, W is the device width, V_{ds} is the drain to source voltage, n is the drain induced barrier lowering (DIBL) coefficient, and $\mu_0 \cdot \mathcal{C}_{ox}$ are constants. Term b in the equation is a technology dependent coefficient obtained empirically.

As discussed previously, leakage energy consumption affected by the miss rate. A miss could potentially increase the program execution time and therefore the circuit would leak for a longer period of time.

Using PTM V1.0[2] and considering the cache to be operating at 360 °K, the total leakage of our proposed cache architecture at different voltage levels for 32- and 45-nm technologies is compared to the leakage of traditional cache architecture and is illustrated in Fig. 7. At nominal voltage the leakage saving compared to a traditional cache (with no IDC and BLB) is negative because of the additional leakage overhead from IDC cache and BLB*.[4] However, at lower voltages, saving in the leakage of cache far outweighs the extra leakage expensed. Notice that only the voltage in the main cache is scaled while the BLB and

[4] It is also possible to completely shut off those components when operating at nominal Vdd therefore eliminating their leakage using power supply gating transistors.

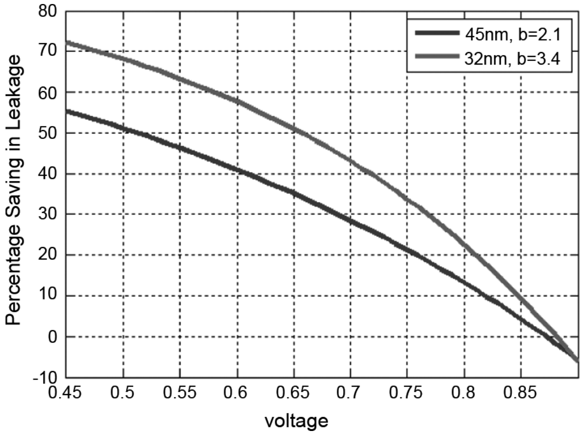

Fig. 7. Percentage saving in the leakage power in proposed architecture compared with the traditional cache architecture.

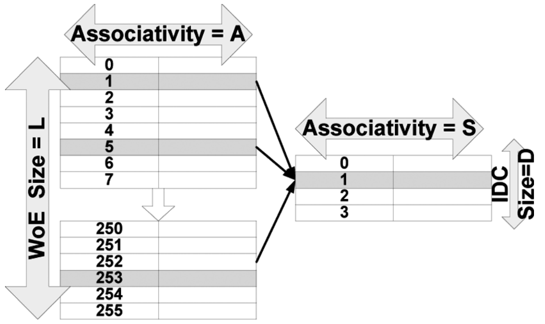

Fig. 8. Sequential mapping of the cache to the IDC.

IDC are kept at their nominal voltages. The leakage savings in the 32 nm grows more rapidly than the 45-nm technology.

VII. Mathematical Bound on IDC Coverage

In this section the upper and lower bounds on the fault tolerance capabilities of IDC is mathematically analyzed. In Section VIII, it will be illustrated that some benchmarks utilize the IDC better than others. The degree of benchmark locality, size of the WoE, and sequentiality of benchmark contribute to the extent of IDC utilization. Considering uniform distribution of MILPV defects in the cache, programs with smaller windows of execution, could better utilize the IDC since expected number of defects in WoE is lower and possibly of overloading IDC and/or addressing many defects to the same IDC location is reduced. In addition, the higher the locality of the information, the higher the chances that a previously mapped fault to IDC is reused before it is overwritten. Finally, for sequential code (codes where the addressing sequence of their program binary within the WoE are sequential (as opposed to hyper branched)) there is uniform chance of mapping conflict across all rows in IDC improving the overall utilization of IDC. This is because as illustrated in Fig. 8, only a subset of locations in the cache could be mapped to each location in IDC.

In order to demonstrate two extremes of fault tolerance performance, we consider two program cases: one with hyper branched addressing (almost every instruction is a branch) and the other with sequential program binary addressing. The probability of a fetch group failure is

$$P_{FG} = 1 - (1 - P_f)^{W.T}. \tag{7}$$

We define a clean access as "an access to one line in the WoE which is either mapped to a defect-free FG in the cache or to a FG previously mapped to the IDC." Note that such access could still result in a miss however the miss is not due to addressing a defective FG, but rather to a failure in tag matching (i.e., the intended information is not in the cache).

Based on Fig. 8, if the binary in the window of execution is highly sequential, LA/D lines in WoE could be mapped to each IDC line. In this case, access to one line is clean if either the accessed FG is defect free , or if it is defective and mapped to IDC and its IDC address is not in conflict with more than "S-1" FGs that need to be mapped to one line in the IDC. In another words, there is less or equal to $S-1$ defects in the other LA/D lines that are mapped to the same physical address in the IDC that the current FG need to be mapped to. Thus, we have

$$P_{\text{clean}}^{\text{access}} = P_{FG} \cdot P_{\text{cov}} + (1 - P_{FG}). \tag{8}$$

In which P_{cov} is the probability that there are less that S other lines mapped to the associated line in IDC

$$P_{\text{cov}} = \sum_{0}^{S-1} P_{\text{mapped}}(i). \tag{9}$$

In which $P_{\text{mapped}}(i)$ stands for the probability that "i" other FGs in the WoE are mapped to that exact same location in IDC. Therefore

$$P_{\text{Mapped}}(i) = \sum_{i=0}^{S-1} \binom{\frac{LA}{D}}{i} (P_{FG})^i \cdot (1 - P_{FG})^{LA/D^{-i}}. \tag{10}$$

For the case of hyper branched binary addressing of the program within the WoE, each of the lines in the window of execution is equally likely to be mapped to any line in the IDC. Therefore the probability of a clean access to one line is obtained by replacing in (10) by LA.

The probability that every access to the WoE is clean is determined by $(P_{clean\_access})^L$. Therefore the probability of clean access in WoE for sequential and hyper branched addressing are

$$P_{\text{cov}}^{\text{sequential}} = \left[\sum_{i=0}^{S-1} \binom{\frac{LA}{D}}{i} (P_{FG})^i \cdot (1 - P_{FG})^{LA/D^{-i}}\right]^L \tag{11}$$

$$P_{\text{cov}}^{\text{Hyper}} = \left[\sum_{i=0}^{S-1} \binom{LA}{i} (P_{FG})^i \cdot (1 - P_{FG})^{LA-i}\right]^L. \tag{12}$$

Based on this analysis it is interesting to investigate how the program behavior in terms of sequentiality could result in different utilization of the IDC. Fig. 9 illustrates the probability of clean access to the WoE for two programs with identical length in WoE utilizing the same size IDC but one with sequential execution in the WoE, and the other with hyper branched binary addressing. One can clearly identify that a program with sequential binary addressing maintains perfectly clean access even for

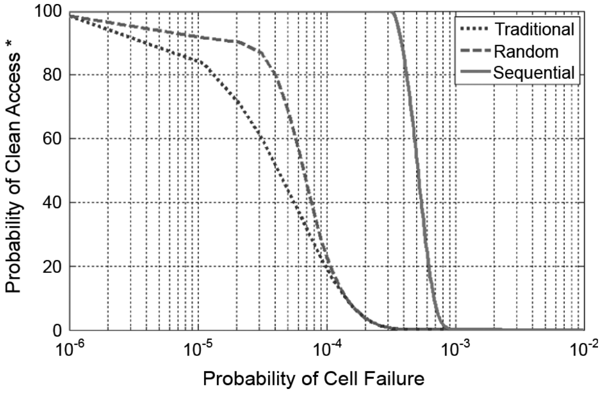

Fig. 9. Probability of clean access for a complete sequential and hyper branched access to the cache when IDC structure is used, compared with probability of clean access for the cache of the same size when IDC is not in use.

$P_f > 10^{-4}$ (and therefore could go to lower Vdd) while a program with hyper branched binary addressing starts to deteriorate much earlier (i.e., at higher Vdd). For practical benchmarks usually the probability of clean access is between these two extremes.

Fig. 10 reveals the effect of changing the size or associatively of the IDC on the probability of clean access. The WoE in all cases is fixed and a local and sequential binary addressing in the WoE is considered. From Fig. 11, it implies that increasing the associativity results in better coverage. However increasing associativity results in increased dynamic power consumption due to the increase in the number of active comparisons in each access. On the other hand, increasing associativity, potentially reduces soft misses (due to access to a defective location) and therefore total energy consumption could improve, which implies an optimal point for the associativity of the IDC. In other words, there exists a pair of (associativity, row size) for each program that will result in the lowest power consumption. However, since for most programs WoE behavior changes dynamically and different programs have different size WoEs it is not possible to find one optimal solution for all scenarios. However, the system can dynamically change the size and associativity of the IDC to target this optimal operating point. Another interesting observation is the relation between the size of the WoE and probability of clean access, or the ability of the IDC to provide a defect free view of the WoE for the processor. Fig. 11 illustrates this relationship for a sequential program.

VIII. Simulation Results

A. Simulation Setup

We used SimpleScalar[1] as a system simulator to study how using IDC improves MILPV tolerance. SimpleScalar is augmented with necessary code to model an IDC-enabled cache hierarchy. The energy consumption of IDC, Cache, and BLB is obtained from post layout simulation of different access scenarios in 32-nm technology. SimpleScalar is updated with this information for the book keeping purpose to calculate the energy consumption based on a per access scenario. The simulated system configuration data is provided in Table VII.

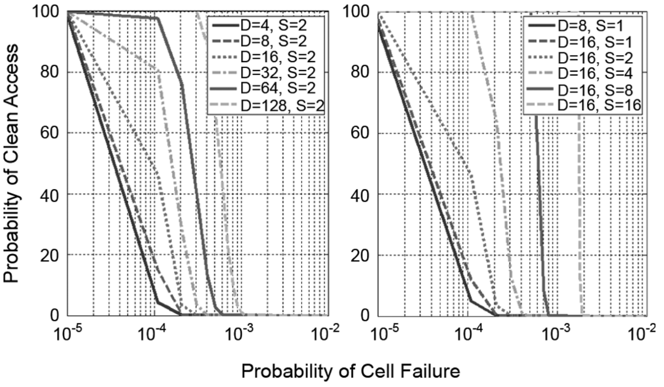

Fig. 10. Comparison of probability of clean access when changing the cache size and associativity.

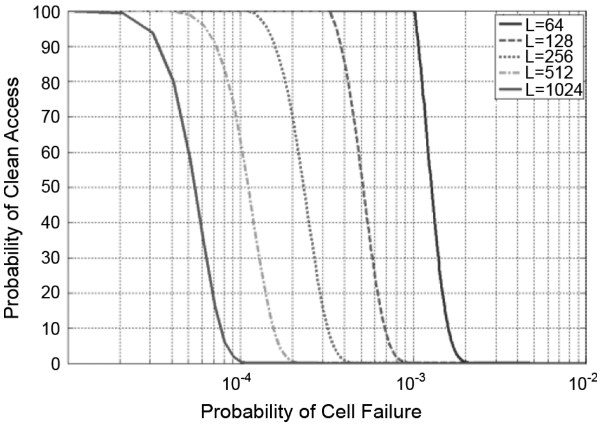

Fig. 11. Probability of clean access for different sizes of WoE.

The cycle time of the cache structures L1 and L2, based on (2) is defined using parameters $(a, b, c) = (1.5, 0, 60)$ at each voltage point. The associated probability of cell failure curve for this simulation is drawn in Fig. 2. In order to calculate the energy consumptions, based on failure probability curves, we assumed uniformly distributed defects in cache structures. After randomly fast forwarding through 1B to 2B instructions in the benchmark, we ran the benchmark for a total of 1B instructions, calculating the sum of dynamic and leakage energy consumption. The process of distribution of defects and the benchmark simulation is repeated 250 times to reduce the chances of accidental over or under utilization.

B. Benchmark Simulation Results

For our case study, we consider the total energy consumption in L1 data and instruction caches as well as the IDC of data and instruction cache and their associated defect map. Figs. 12 and 13 illustrate the percentage of energy saving of each benchmark in data and instruction cache respectively when compared to its total energy consumption at the nominal voltage. It is assumed that the IDC and BLB are power gated when the whole cache is operating at nominal voltage in order to eliminate their leakage.

The savings in dynamic energy consumption increases when reducing the supply voltage but only up to a certain point referred to as "turning point," or V_{TP}. Further lowering the $V_{dd} <$

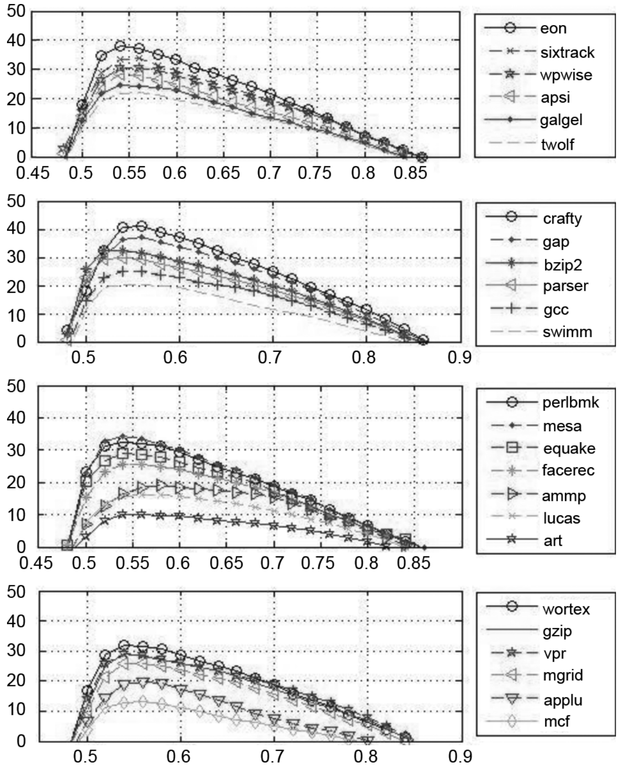

Fig. 12. Percentage saving in total energy consumption for different SPEC2000 benchmarks in L1 data cache.

V_{TP} increases the number of defective FGs beyond the tolerance of the proposed architecture and abruptly increases the miss rate, sharply increasing the dynamic energy consumption. In addition to the supply voltage, execution properties of each benchmark affects the extent of its energy saving. IDC stores defective FGs inside the WoE. Therefore programs with rapidly changing WoE (multiple branching, and generally less local), or those having very large window (such that the number of defective words inside that window exceeds the IDC size), will gain lower energy savings. The "eon" and "crafty" benchmark in Figs. 12 and 13 are programs with multiple small but long executing loops, small execution window and good locality. On the other extreme, "mcf" and "art" benchmark are less local with rapidly changing execution windows. Other benchmarks fall between these two extremes. Note that when migrating to more aggressive nodes (smaller geometries), leakage will become dominant, allowing lower V_{TP}. However, there will always be a saturation point since the extra soft misses not only increase the dynamic power consumption by requiring access to lower level memories, but also increase the execution time allowing the circuit to leak for a longer period.

The energy savings that are reported in Figs. 12 and 13 are for a VFS policy defined by parameters $(a,\ b,\ c) = (1.5,\ 0,\ 60)$. By choosing a different set of parameters for (a,b,c) the failure rate and therefore the amount of utilization of IDC at different voltages will change. Table IV reports the best, worst and average power savings as well as V_{TP} associated with different policies (32-nm technology). As a reminder, when a and b are chosen 0, the voltage scaling policy is FFVS.

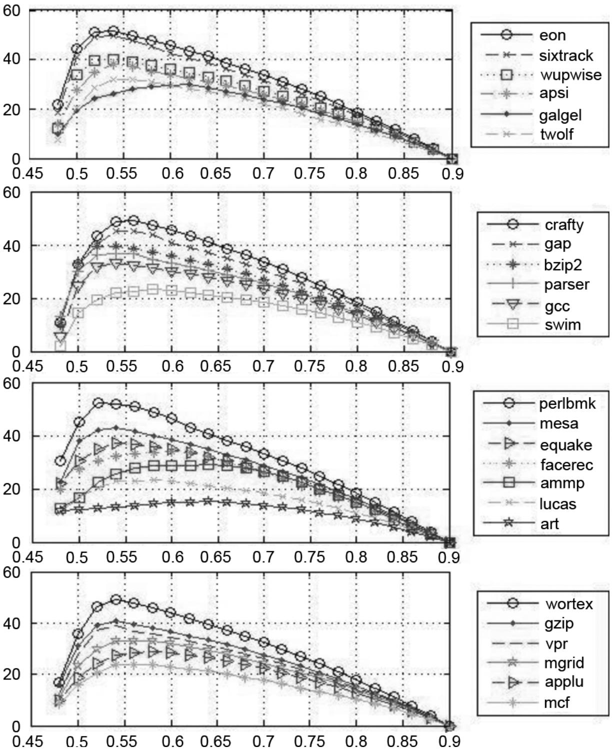

Fig. 13. Percentage saving in total energy consumption for different SPEC2000 benchmarks in L1 instruction Cache.

TABLE IV
MIN/MAX/AVERAGE VOLTAGE TURNING POINTS AND ENERGY SAVING ACROSS ALL BENCHMARK IN 32 nm

Policy, (a,b,c)	Best energy savings (%), $V_{TP}(V)$	Worse energy savings (%), $V_{TP}(V)$	Average energy savings (%), $V_{TP}(V)$
VFS(0.5,2,50)	36.70, 0.74	18.61, 0.83	27.11, 0.8
VFS(0,2,100)	48.21,0.52	18.54,0.63	33.92,0.53
FFVS(0,0,180)	26.92,0.65	16.13,0.69	21.71, 0.67
FFVS(0,0,220)	36.56,0.6	21.15,0.64	30.01,0.62
FFVS(0,0,260)	44.16,0.54	21.63,0.56	32.12,0.55

C. Choosing the IDC Size

By choosing different sizes of IDC, one could trade area for larger fault tolerance and consequently change the amount of expected power savings. Using a larger size IDC reduces the number of soft misses but at the same time increases static power. Depending on its organization, a larger IDC could also consume larger dynamic energy. We now attempt to quantify those tradeoffs using area and power savings estimates based on estimation and simulation approach.

1) Area Overhead Analysis: Using area figures from CACTI for the main cache (16_{KB}, 2 Banks), IDC (of size 32 rows /2 way associative) and BLB in 32-nm technology, an area overhead of 7.69% is incurred. This area overhead is less than that reported in [3] since the defect map is generated at a larger granularity (FG rather than word) and therefore the area of BLB is reduced. Our layout of an IDC enabled cache (in 45-nm technology using TSMC-45LG and ICstudios) resulted in a design with 7.31% area overhead. The total area penalty (including BLB and IDC)

TABLE V
TOTAL AREA PENALTY OF L1 INSTRUCTION CACHE WHEN USING DIFFERENT IDC SIZES (2-WAY CACHE IS ASSUME)

IDC Size (rows)	4	8	16	32	64
Area Penalty (%)	4.81	5.77	7.69	11.54	19.24

for different choices of IDC size in 32-nm technology node is summarized in Table V.

Note that based on the size of the cache, the technology, the size ratio of IDC to cache, and many other parameters, this area overhead is subject to change. However our purpose in providing the area overhead data is to clarify that the area cost to implement the proposed cache is indeed small and acceptable. Furthermore, it is important to note that the proposed scheme can also be used to correct for manufacturing defects and therefore remove the need for built-in self repair (BISR) (but not BIST), the actual penalty can be much less when compared to memories with BISR.

2) Power Saving Analysis: In order to illustrate how choosing the IDC size changes the expected power saving, we simulated the configuration given in Table III for IDC sizes of (4, 8, 16, 32, and 64) across three different benchmarks. Benchmark "perlbmk" represent the highly localized benchmark that well utilizes the IDC. On the other hand benchmark "lucas" represent the opposite extreme benchmarks with less localized addressing pattern and finally benchmark "equake" that represent the average case benchmarks.

Fig. 14 illustrates the energy savings for the above benchmarks as a function of V_{dd} and IDC size. A Noticeable trend is the difference in the expected energy savings of each bench mark for different sizes of IDC at lower voltages. At lower voltages, the smaller IDCs consume less static power and therefore result in better energy savings. However the smaller size IDCs reach the V_{TP} earlier than larger size IDCs; therefore their maximum power savings could be less than that of a larger IDCs. As the IDC size grows, the maximum power saving also grows but the difference with the previous point become smaller, and at some point (IDC size = 64 in average and low benchmark) even the larger size IDC is expected to have equal or less power saving compare to a smaller IDC size (due to large increase in the probability of failure at lower voltages and also increased static power saving in the larger size IDCs).

IX. USING IDC TO REALIZE A FASTER MACHINE

Basically IDC increases the fault tolerance of the Cache from that handled by redundancy alone to a much higher failure rate. On the other hand based on what was discussed before regarding the distribution of access and write time, we have demonstrated that by shortening the cycle time the probability of failure increases. An alternative way to utilize the IDC is to realize the fastest fault-free machine at a given supply voltage by shortening the cycle time up to the limit of IDC fault handling capability. This effectively defines the upper limit of a VFS policy at each voltage point.

In order to demonstrate the idea for the system defined in Table III and using SimpleScalar[1] we simulated the system for different random distribution of faults in the L1 data and instruction caches. As it turned out for probability of failure of 6×10^{-4} or less the IDC is capable of hiding the defect from the processor for all the SPEC2000 benchmarks. This probability of failure for some other choices of IDC size is listed in Table VI. Knowing the highest tolerable failure probability of the system, we searched at each voltage point (trough a Monte Carlo simulation) for an access time that provides such probability of failure.

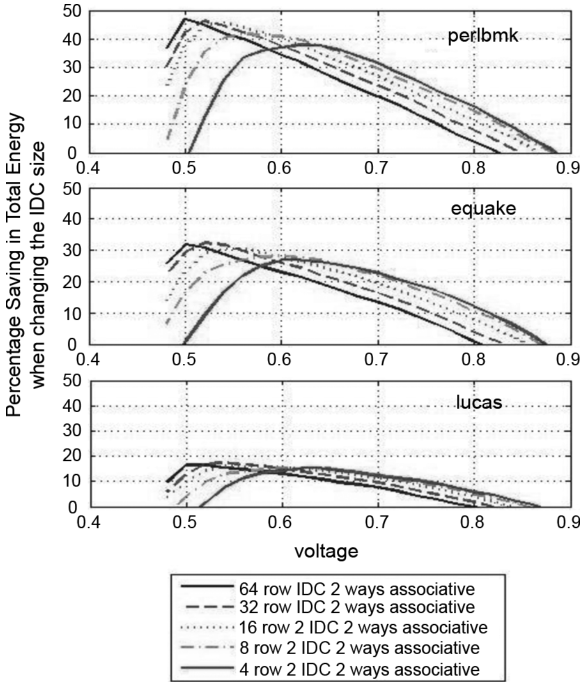

Fig. 14. Percentage saving in total energy consumption for perlbmk, equake, and lucas benchmarks in L1 data cache when changing the IDC size.

TABLE VI
TOLERABLE CELL FAILURE PROBABILITY FOR DIFFERENT SIZES OF IDC VALID FOR ALL SPEC2000 BENCHMARKS FOR A $16_{\rm KB}$L1 CACHE

IDC Size	2 Row 2 Way	4 Row 2 Way	8 Row 2 Way	16 Row 2 Way	32 Row 2 Way
Pf	7.54E-5	1.45E-4	2.64E-4	6.02E-4	1.24E-3

In order to define a baseline, the same analysis is performed for a cache without IDC. In this case, the cycle time should be large enough such that the resulting probability of failure achieves the production yield expectation, meaning it is small enough that it could be tolerated using available redundancy. Consider, as an example, that the probability of failure of 10^{-6} is desired to meet the expected yield for a $16_{\rm KB}$ cache. Through a Monte Carlo simulation for a 10^{-6} failure rate, one can obtain the associated cycle time. Fig. 15 compares the minimum cycle time of a $16_{\rm KB}$ IDC to that of a traditional cache of the same size. As illustrated, the IDC allows the system to operate at lower cycle times that are can be up to 25% shorter at sub-500 mV supply.

X. EFFECT OF CHANGE IN THRESHOLD VOLTAGE DISTRIBUTION

Threshold voltage variation is the cumulative result of variation in the physical parameters of transistors devices. The major contributors to the threshold voltage variation in sub

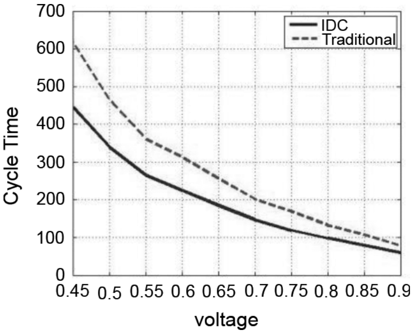

Fig. 15. Minimum cycle time improvement when IDC is used to realize a faster machine.

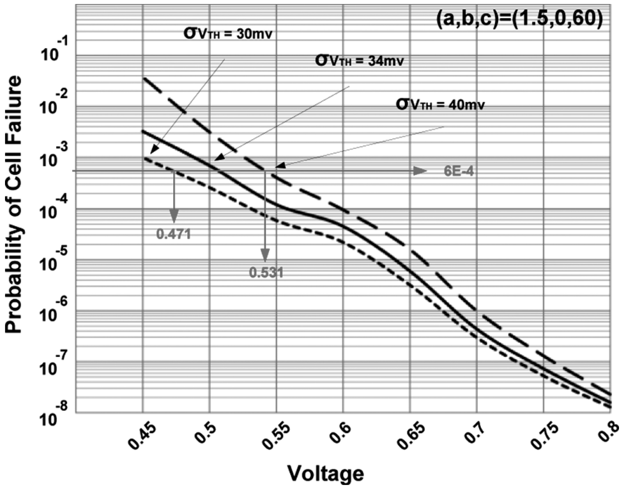

Fig. 16. Probability of cell failure for different standard deviation of threshold voltage (σV_{TH}).

100_{nm} manufacturing, as discussed in Section III, are the RDF and LER. The degree of introduced variation changes from factory to factory. In this section we examine the change in the probability of memory cell failure for different measures of threshold voltage standard deviation (σV_{TH}). Fig. 16 illustrates the probability of memory cell failure (P_F) obtained from Monte Carlo simulation when the standard deviation of threshold voltage (σV_{TH}) is chosen form (30, 34, 40 mV) accounting for more and less matured processes. In order to be fair the failure probabilities are calculated for memories with the same performance (cycle time); the memory cycle time at each voltage is constant across different choices of σV_{TH} and is determined based on (2) using parameters $(a, b, c) = (2, 0, 60)$ for the $\sigma V_{TH} = 34_{\mathrm{mV}}$. The choice of $(\mathrm{b} = 0)$ in (2) removes the correlation of memory cycle time to the threshold voltage distribution, allowing us to study the trend of change in the failure probability with change in standard deviation of threshold voltage. Fig. 16 illustrates that with reduction in σV_{TH} the probability of cell failure across different voltages improves. In addition the change in the threshold voltage affects the P_F more severely in lower voltages. Configuring a 16 rows long two ways associative IDC with a 16_{KB} L1 cache, based on Table VI, successfully hides defect rates below 6×10^{-4} with negligible loss in performance. This probability of failure, as suggested by Fig. 16, is obtained at a different voltage as the (σV_{TH}) changes; a more mature process exhibit the same probability of failure at a lower voltage, allowing the 16_{KB} cache to be operated at a lower voltage (V_{cc}^{Min}). Table VII summarizes the V_{cc}^{Min} for different standard deviation when probability of cell failure is kept constant on 6×10^{-4} while $(a, b, c) = (1.5, 0, 60)$ equation is used for cycle time. Reduced σV_{TH} allows a lower (V_{cc}^{Min}). Ability to use a lower (V_{cc}^{Min}) improves the extent of achievable energy saving. To better illustrate this we have simulated the "crafty" benchmark (from SPEC2000 suit) across range of defined by change in σV_{TH} for a fixed failure probability of 6×10^{-4}. System simulation is performed 300 times. In each simulation defects are randomly and uniformly distributed in the memory, the system is then simulated and based on book keeping data the energy consumption of the system for each choice of $(V_{cc}^{\mathrm{Min}}, \sigma V_{TH})$ is calculated and compared with that of $\sigma V_{TH} = 34_{\mathrm{mV}}$. Change in the percentage of energy consumption in comparison with that of $\sigma V_{TH} = 34_{\mathrm{mV}}$ is referred to as Δ, calculated based on (13) and presented in Table VII

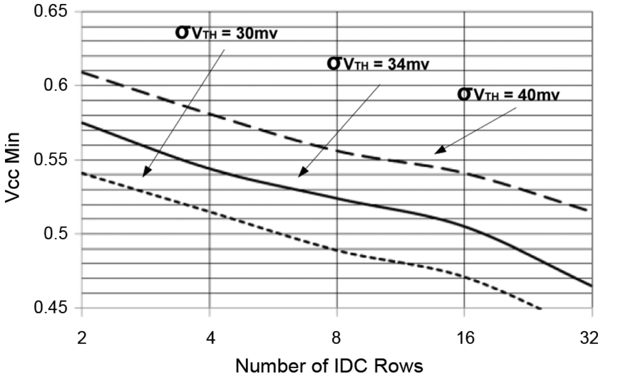

Fig. 17. Change in the minimum voltage with change number of IDC rows with associativity of 2.

TABLE VII
EFFECT OF CHANGE IN $\sigma_{V_{th}}$ ON V_{cc}^{Min} AND TOTAL ENERGY CONSUMPTION, ENERGY SAVINGS OF THE SYSTEM FOR DIFFERENT $\sigma_{V_{th}}$ IS COMPARED WITH $\sigma = 34_{\mathrm{mv}}$ CASE AND IS REPRESENTED BY Δ

$\sigma_{V_{th}}$	30	34	40
V_{cc}^{Min}	0.471	0.505	0.541
Δ	+2.128	0	-2.867

$$\Delta_X = \frac{100 * (E_{V_{TH=X}} - E_{V_{TH=34\ \mathrm{mV}}})}{E_{V_{TH=34\ \mathrm{mV}}}}. \qquad (13)$$

IDC size determines the extent of system's fault tolerance. Choosing a larger IDC increase the area overhead, in turn reducing the C_{cc}^{Min} by tolerating more defects. Fig. 17 illustrates how changing the size of IDC changes the limit of the C_{cc}^{Min}. This figure also captures the effect of the variation in the standard deviation of threshold voltage on the limit of C_{cc}^{Min}.

XI. CONCLUSION

In this paper, we targeted caches for aggressive supply voltage scaling. We present a new architecture that uses an auxiliary cache (IDC) and a defect map (BLB) to enable significant overall power savings while maintaining low failure rates. The proposed approach can be extended in many ways, such as considering whole memory hierarchies, and defining policies for voltage-frequency power management that are more effective than traditional ones assuming "defect free" operation. The impact of this technique becomes more pronounced with decreased process geometries.

REFERENCES

[1] S. Mukhopadhyay, H. Mahmoodi, and K. Roy, "Modeling of failure probability and statistical design of SRAM array for yield enhancement in nanoscaled CMOS," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 24, no. 12, pp. 1859–1880, Dec. 2005.

[2] A. K. Djahromi, A. Eltawil, F. J. Kurdahi, and R. Kanj, "Cross layer error exploitation for aggressive voltage scaling," in *Proc. 8th Int. Symp. Quality Electron. Des. (ISQED)*, Mar. 2007, pp. 192–197.

[3] M. A. Makhzan, A. Khajeh, A. Eltawil, and F. J. Kurdahi, "Limits on voltage scaling for caches utilizing fault tolerant techniques," in *Proc. 25th Int. Conf. Comput. Des., (ICCD)*, Oct. 2007, pp. 488–495.

[4] Hewlett-Packard Laboratories, Palo Alto, CA, "CACTI ; An integrated cache and memory access time, cycle time, area, leakage, and dynamic power model," 2010. [Online]. Available: http://www.hpl.hp.com/personal/Norman_Jouppi/cacti4.html

[5] P. Shirvani and E. J. McCluskey, "PADded cache: A new fault-tolerance technique for cache memories," in *Proc. 17th IEEE VLSI Test Symp.*, 1999, pp. 440–445.

[6] H. T. Vergos and D. Nikolos, "Efficient fault tolerant cache memory design," *Microprocess. Microprogram. J.*, vol. 41, no. 2, pp. 153–169, 1995.

[7] G. S. Sohi, "Cache memory organization to enhance the yield of high performance VLSI processors," *IEEE Trans. Comput.*, vol. 38, no. 4, pp. 484–492, Apr. 1989.

[8] A. F. Pour and M. D. Hill, "Performance implications of tolerating cache faults," *IEEE Trans. Comput.*, vol. 42, no. 3, pp. 257–267, Mar. 1993.

[9] L. Xiao and J. C. Muzio, "A fault-tolerant multiprocessor cache memory," in *Proc. Rec. IEEE Int. Workshop Memory Technol., Des., Test.*, Aug. 1994, pp. 52–57.

[10] Y. Ooi, M. Kashimura, H. Takeuchi, and E. Kawamura, "Fault-tolerant architecture in a cache memory control LSI," *IEEE J. Solid-State Circuits*, vol. 27, no. 4, pp. 507–514, Apr. 1992.

[11] A. Agarwal, B. C. Paul, S. Mukhopadhyay, and K. Roy, "Process variation in embedded memories: Failure analysis and variation aware architecture," *IEEE J. Solid-State Circuits*, vol. 40, no. 9, pp. 1804–1814, Sep. 2005.

[12] C. Wilkerson, G. Hongliang, A. R. Alameldeen, Z. Chishti, M. Khellah, and L. Shih-Lien, "Trading off Cache Capacity for Reliability to Enable Low Voltage Operation," in *Proc. Comput. Arch.*, Jun. 2008, pp. 203–214.

[13] A. J. Bhavnagarwala, T. Xinghai, and J. D. Meindl, "The impact of intrinsic device fluctuations on CMOS SRAM cell stability," *IEEE J. Solid-State Circuits*, vol. 36, no. 4, pp. 658–665, Apr. 2001.

[14] A. Agarwal, K. Roy, and T. N. Vijaykumar, "Exploring high bandwidth pipelined cache architecture for scaled technology," in *Proc. Des., Autom., Test Euro. Conf. Exhibition*, 2003, pp. 778–783.

[15] M. A. Makhzan, A. Khajeh, A. Eltawil, and F. J. Kurdahi, "A low power JPEG2000 encoder with iterative and fault tolerant error concealment," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 17, no. 6, pp. 827–837, Jun. 2009.

[16] K. Flautner, K. N. Sung, S. Martin, D. Blaauw, and T. Mudge, "Drowsy caches: Simple techniques for reducing leakage power," in *Proc. 29th Annu. Int. Symp. Comput. Arch.*, 2002, pp. 148–157.

[17] H. Homayoun, A. Sasan, and A.V. Veidenbaum, "Multiple sleep mode leakage control for cache peripheral circuits in embedded processors," in *Proc. CASES*, 2008, pp. 197–206.

[18] M. A. Lucente, C. H. Harris, and R. M. Muir, "Memory system reliability improvement through associative cache redundancy," in *Proc. IEEE Custom Integr. Circuits Conf.*, May 1990, pp. 19.6/1–19.6/4.

[19] Z. Bo, D. Blaauw, D. Sylvester, and K. Flautner, "The limit of dynamic voltage scaling and insomniac dynamic voltage scaling," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 13, no. 11, pp. 1239–1252, Nov. 2005.

[20] X. Tang, V. K. De, and J. D. Meindl, "Intrinsic MOSFET parameter fluctuations due to random dopant placement," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 5, no. 4, pp. 369–376, Dec. 1997.

[21] K. Ishimaru, "45 nm/32 nm CMOS challenge and perspective," in *Proc. 37th Euro. Solid State Device Res. Conf. (ESSDERC)*, Sep. 2007, pp. 32–35.

[22] R. K. Krishnarnurthy, A. Alvandpour, V. De, and S. Borkar, "High-performance and low-power challenges for sub-70 nm microprocessor circuits," in *Proc. IEEE Custom Integr. Circuits Conf.*, 2002, pp. 125–128.

Avesta Sasan (S'05) received the B.S. degree (*summa cum laude*) in computer engineering and the M.S. and Ph.D. degrees in electrical engineering from University of California, Irvine, in 2005, 2006, and 2010, respectively.

His research interests include low power design, process variation aware architectures, fault tolerant computing systems, nano-electronic power and device modeling, VLSI signal processing, processor power and reliability optimization, and logic-architecture-device codesign. His latest publication, research outcomes and updates could be found on http://www.avestasasan.com.

Houman Homayoun received the B.S. degree in electrical engineering from Sharif University of Technology, Tehran, Iran, in 2003, the M.S. degree in computer engineering from University of Victoria, Victoria, BC, Canada, in 2005, and the Ph.D. degree in computer science from University of California, Irvine, in 2010.

He is with the Department of Electrical Engineering and Computer Science, University of California, Irvine. His research is on power-temperature and reliability-aware memory and processor design optimizations and spans the areas of computer architecture and VLSI circuit design. His research is among the first in the field to address the importance of cross-layer power and temperature optimization in memory peripheral circuits. The results of his research were published in top-rated conferences including ISLPED, DAC, DATE, HiPEAC, CASES, ICCD, CF , and LCTES.

Dr. Homayoun was the recipient of the 4-years University of California Irvine School of Information and Computer Science Chair Fellowship.

Ahmed M. Eltawil (M'97) received the Doctorate degree from the University of California, Los Angeles, in 2003.

He is an Associate Professor with the University of California, Irvine, where he has been with the Department of Electrical Engineering and Computer Science since 2005. He is the founder and director of the Wireless Systems and Circuits Laboratory (http://newport.eecs.uci.edu/~aeltawil/). In 2010, he holds a visiting professorship with King Saud University, Saudi Arabia. His current research interests include low power digital circuit and signal processing architectures for wireless communication systems with a focus on physical layer design where he has published over 50 technical papers on the subject, including four book chapters.

Dr. Eltawil was a recipient of several distinguished awards, including the NSF CAREER Award in 2010. Since 2006, he has been a member of the Association of Public Safety Communications Officials (APCO) and has been actively involved in efforts towards integrating cognitive and software defined radio technology in critical first responder communication networks.

Fadi J. Kurdahi (M'87–SM'03–F'05) received the Ph.D. degree from the University of Southern California, Los Angeles, in 1987.

Since 1987, he has been a faculty member with the Department of Electrical and Computer Engineering, University of California, Irvine (UCI), where he conducts research in the areas of computer-aided design of VLSI circuits, high-level synthesis, and design methodology of large scale systems, and serves as the Associate Director for the Center for Embedded Computer Systems (CECS).

Dr. Kurdahi was an Associate Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—PART II: BRIEF PAPERS 1993–1995, Area Editor in IEEE DESIGN AND TEST for reconfigurable computing, and served as program chair, general chair, or on program committees of several workshops, symposia and conferences in the area of CAD, VLSI, and system design. He was a recipient of the Best Paper Award for the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS in 2002, the Best Paper Award in 2006 at ISQED, four other Distinguished Paper Awards at DAC, EuroDAC, ASP- DAC, and ISQED, and the Distinguished Alumnus Award from this Alma Mater, the American University of Beirut in 2008. He is a Fellow of the AAAS.